Supplementary File - Data preprocessing to mitigate bias: A maximum entropy based approach

A. Properties of the reweighting algorithm

The prior distribution we construct has the following form. For $C \in [0,1]$,

$$q_C^w(\alpha) = C \cdot u(\alpha) + (1-C) \cdot w(\alpha). \tag{1}$$

Here u is the uniform distribution over ω. The weight distribution w is obtained using Algorithm 1 to satisfy certain statistical rate constraints. To establish the correctness of Algorithm 1, we can estimate the representation rate and statistical rate of this distribution and formally prove Theorem 4.1.

Theorem A.1 (Theorem 4.1). *There is an algorithm (Algorithm 1) that, given the dataset $\mathcal{S}$ and a $\tau \in [0,1]$, outputs a probability distribution $w : \mathcal{S} \to [0,1]$ such that*

1. *The algorithm runs in time linear in N.*
2. *w satisfies τ-statistical rate.*

As a consequence, q_C^w defined in (1) also satisfies τ-statistical rate.

To prove this theorem, we will consider the uniform and weighted part of the q_C^w separately and show that the convex combination of two distributions satisfies similar fairness properties as the two distributions. We start with the statements and proofs of bounds for the uniform distribution.

Lemma A.2. *Let $u : \Omega \to [0,1]$ be the uniform distribution on Ω. Then u satisfies the following properties.*

1. *For a fixed $y \in \mathcal{Y}$,*
$$u(Y = y, Z = 0) = u(Y = y, Z = 1).$$
2. $u(Z = 0) = u(Z = 1)$.
3. *For a fixed $y \in \mathcal{Y}$,*
$$u(Y = y \mid Z = 0) = u(Y = y \mid Z = 1).$$

Proof. (1) For any $\alpha \in \Omega$, let $y(\alpha)$ denote the class label of element α and let $z(\alpha)$ denote the sensitive attribute value of element α.

$$\begin{aligned} u(Y = y, Z = z) &= \sum_{\alpha \in \Omega \ | y(\alpha) = y, z(\alpha) = z} \frac{1}{|\Omega|} \\ &= \frac{1}{|\Omega|} \cdot \frac{|\Omega|}{2|\mathcal{Y}|} = \frac{1}{2|\mathcal{Y}|}. \end{aligned}$$

Since the above term is independent of z-value, $u(Y = y, Z = z)$ is equal, for all z.

(2) Using

$$u(Z = z_1) = \sum_{y \in \mathcal{Y}} u(Z = z_1, Y = y).$$

and part (1), we get

$$\sum_{y \in \mathcal{Y}} u(Z = z_1, Y = y) = \sum_{y \in \mathcal{Y}} u(Z = z_2, Y = y).$$

This implies that

$$u(Z = z_1) = u(Z = z_2).$$

(3) Taking the ratio of part (1) and (2), we get

$$\begin{aligned} u(Y = y \mid Z = z_1) &= \frac{u(Y = y, Z = z_1)}{u(Z = z_1)} \\ &= \frac{u(Y = y, Z = z_2)}{u(Z = z_2)} \\ &= u(Y = y \mid Z = z_2). \end{aligned}$$

□

As expected, the uniform distribution is perfectly fair. We next try to prove similar bounds for the weighted distribution w.

Lemma A.3. *Given dataset $\mathcal{S}$ and parameter $\tau \in [0,1]$, let w be the weighted distribution on samples in $\mathcal{S}$ obtained from Algorithm 1 with input $\mathcal{S}$ and τ. Then w satisfies the following properties.*

1. *For a fixed $y \in \mathcal{Y}$,*
$$w(Y = y, Z = 0) = \tau \cdot w(Y = y, Z = 1).$$
2. $w(Z = 0) = \tau \cdot w(Z = 1)$.
3. *For a fixed $y \in \mathcal{Y}$,*
$$w(Y = y \mid Z = 0) = w(Y = y \mid Z = 1).$$

Proof. Note that, by definition, the support of w is the elements in the dataset $\mathcal{S}$. For any $\alpha \in \Omega$, let $y(\alpha)$ denote the class label of element α and let $z(\alpha)$ denote the sensitive attribute value of element α.

(1) For any value $z \in \{0, 1\}$,

$$w(Z = z, Y = y) = \sum_{\alpha \in S \ | y(\alpha)=y, z(\alpha)=z} w(\alpha)$$

We will analyze the elements with sensitive attribute value 0 and 1 separately since they have different weights. From Algorithm 1,

$$\begin{aligned} w(Y = y, Z = 1) &= \sum_{\alpha \in S \ | y(\alpha)=y, z(\alpha)=1} w(\alpha) \\ &= \sum_{\alpha \in S \ | y(\alpha)=y, z(\alpha)=1} \frac{1}{W} \sum_{i=1}^{N} \mathbb{1}(\alpha_i = \alpha) \cdot \frac{c(y)}{c(y, 1)} \\ &= \frac{1}{W} \cdot c(y, 1) \cdot \frac{c(y)}{c(y, 1)} = \frac{c(y)}{W}. \end{aligned}$$

Similarly, for elements with sensitive attribute value 0,

$$\begin{aligned} w(Y = y, Z = 0) &= \sum_{\alpha \in S \ | y(\alpha)=y, z(\alpha)=0} w(\alpha) \\ &= \sum_{\alpha \in S \ | y(\alpha)=y, z(\alpha)=0} \frac{1}{W} \sum_{i=1}^{N} \mathbb{1}(\alpha_i = \alpha) \cdot \frac{\tau \cdot c(y)}{c(y, 0)} \\ &= \frac{1}{W} \cdot c(y, 0) \cdot \frac{c(y)}{c(y, 0)} = \frac{\tau \cdot c(y)}{W}. \end{aligned}$$

Therefore,

$$\frac{w(Y = y, Z = 0)}{w(Y = y, Z = 1)} = \tau \text{ and } \frac{w(Y = y, Z = 1)}{w(Y = y, Z = 0)} = \frac{1}{\tau} \geq 1.$$

Hence, the ratio for z_1, z_2 is atleast τ.

(2) The statement of part (1) holds for all $y \in \mathcal{Y}$. Therefore,

$$\sum_{y \in \mathcal{Y}} w(Z = z_1, Y = y) \geq \tau \cdot \sum_{y \in \mathcal{Y}} w(Z = z_2, Y = y).$$

This implies that

$$w(Z = z_1) \geq \tau \cdot w(Z = z_2).$$

Since the probability mass assigned to all sensitive attribute values are within a τ-factor of each other, the representation rate of w is atleast τ. In particular, using the exact inequalities in the proof of part (1), we get

$$\sum_{y \in \mathcal{Y}} w(Z = 0, Y = y) = \tau \cdot \sum_{y \in \mathcal{Y}} w(Z = 1, Y = y)$$

which implies that

$$w(Z = 0) = \tau \cdot w(Z = 1).$$

(3) Taking the ratio of part (1) and (2), we get

$$w(Y = y \mid Z = 0) = \frac{w(Y = y, Z = 0)}{w(Z = 0)} = w(Y = y \mid Z = 1)$$

□

Before using the above properties of uniform and weighted distribution to prove Theorem 4.1, we will show that the convex combination of two distributions has similar fairness guarantees as the two distributions.

Lemma A.4 (Statistical rate of convex combination of two distributions). *Given distributions v_1, v_2 on domain Ω and a parameter $C \in [0, 1]$, define distribution q as*

$$q(\alpha) := C \cdot v_1(\alpha) + (1 - C) \cdot v_2(\alpha).$$

For parameters for $0 < \tau_2 \leq \tau_1 \leq 1$, suppose that v_1, v_2 satisfy the following properties:

1. $v_1(Z = 0) = \tau_1 \cdot v_1(Z = 1)$ *and* ,
$$v_2(Z = 0) = \tau_2 \cdot v_2(Z = 1).$$
2. *For a fixed $y \in \mathcal{Y}$,*
$$v_1(Y = y, Z = 0) = \tau_1 \cdot v_1(Y = y, Z = 1) \text{ and },$$
$$v_2(Y = y, Z = 0) = \tau_2 \cdot v_2(Y = y, Z = 1).$$

Then for a fixed $y \in \mathcal{Y}$ and $z_1, z_2 \in \{0, 1\}$, q satisfies the following properties

1. $q(Y = y \mid Z = z_1) \geq \tau_1 \tau_2 \cdot q(Y = y \mid Z = z_2)$.
2. $$\frac{q(Y = y, Z = 0)}{q(Y = y, Z = 1)} \geq \tau_2 \text{ and}$$
$$\frac{q(Y = y, Z = 1)}{q(Y = y, Z = 0)} \geq 1.$$

Proof. From the definition of q,

$$q(Z = 0) = C \cdot v_1(Z = 0) + (1 - C) \cdot v_2(Z = 0).$$

Using the first property of v_1 and v_2, we get

$$\begin{aligned} q(Z = 0) &= C \cdot \tau_1 \cdot v_1(Z = 1) + (1 - C) \cdot \tau_2 \cdot v_2(Z = 1) \\ &= \tau_2 \cdot (C \cdot v_1(Z = 1) + (1 - C) \cdot v_2(Z = 1)) \\ &\quad + C \cdot (\tau_1 - \tau_2) \cdot v_1(Z = 1) \\ &= \tau_2 \cdot q(Z = 1) + C \cdot (\tau_1 - \tau_2) \cdot v_1(Z = 1) \\ &\geq \tau_2 \cdot q(Z = 1). \end{aligned}$$

The last inequality holds because $\tau_2 \leq \tau_1$. Similarly, since $\tau \in (0, 1]$,

$$\begin{aligned} q(Z = 1) &= C \frac{1}{\tau_1} \cdot v_1(Z = 0) + (1 - C) \cdot \frac{1}{\tau_2} \cdot v_2(Z = 0) \\ &\geq \frac{1}{\tau_1} \cdot q(Z = 0) + (1 - C) \cdot (\frac{1}{\tau_2} - \frac{1}{\tau_1}) \cdot v_1(Z = 0) \\ &\geq \frac{1}{\tau_1} \cdot q(Z = 0). \end{aligned}$$

In other words, the representation rate of q is atleast τ_2. Once again, using the definition of q,

$$\begin{aligned} q(Y = y, Z = 0) = C \cdot v_1(Y = y, Z = 0) \\ + (1 - C) \cdot v_2(Y = y, Z = 0). \end{aligned}$$

Using the properties of v_1, v_2, we can alternately write the

above expression as

$$q(Y=y,Z=0) = C\cdot\tau_1\cdot v_1(Y=y,Z=1) + (1-C)\cdot\tau_2\cdot v_2(Y=y,Z=1).$$

Let

$$a = C\cdot v_1(Y=y,Z=1) \text{ and}$$

$$b = (1-C)\cdot v_2(Y=y,Z=1).$$

Then,

$$\frac{q(Y=y,Z=0)}{q(Y=y,Z=1)} = \frac{a\tau_1+b\tau_2}{a+b} = \tau_2 + \frac{(\tau_1-\tau_2)a}{a+b} \geq \tau_2,$$

since $a, b, (\tau_1-\tau_2) \geq 0$. Similarly, since $\tau_1, \tau_2 \in [0,1]$

$$\frac{q(Y=y,Z=1)}{q(Y=y,Z=0)} = \frac{a+b}{a\tau_1+b\tau_2} \geq 1.$$

Hence the ratio of the joint distributions for different values of sensitive attributes is atleast τ. Now to prove the statistical rate bound, we just need to take the ratio of the joint distribution and marginal distribution. Taking the ratio we get,

$$\begin{aligned} q(Y=y \mid Z=0) &= \frac{q(Y=y,Z=0)}{q(Z=0)} \\ &\geq \frac{\tau_2\cdot q(Y=y,Z=1)}{\frac{1}{\tau_1}q(Z=1)} \\ &= \tau_1\tau_2\cdot q(Y=y\mid Z=1). \end{aligned}$$

Similarly,

$$\begin{aligned} q(Y=y \mid Z=1) &= \frac{q(Y=y,Z=1)}{q(Z=1)} \\ &\geq \frac{q(Y=y,Z=0)}{\frac{1}{\tau_2}\cdot q(Z=0)} \\ &= \tau_2\cdot q(Y=y\mid Z=0). \end{aligned}$$

Since $\tau_2 \leq \tau_1 \leq 1$, the minimum of the two ratios is $\tau_1\tau_2$. Hence the statistical rate of q is $\tau_1\tau_2$.

□

While the first result of the above lemma bounds the statistical rate of q, the second result will be useful in bounding the statistical rate of the max-entropy distribution obtained using q. Using Lemma A.4, we can now prove the representation rate and statistical rate bound on the prior q_C^w.

Proof of Theorem 4.1. Proving the first statement is simple. Since Algorithm 1 just counts the number of elements in $\mathcal{S}$ satisfying certain properties, the time taken is $|\mathcal{Y}|\cdot N$. In case of hypercube domain, $|\mathcal{Y}| = 2$. Hence the time complexity of the re-weighting algorithm is linear in N.

For the statistical rate of q_C^w, plugging $v_1 = u$ and $v_2 = v^w$ in Lemma A.4, we can get the corresponding ratio for q_C^w. In particular, from Lemma A.2 and Lemma A.3, we know that $\tau_1 = 1$ for distribution u and $\tau_2 = \tau$ for distribution v^w. The statement of Lemma A.4 then tells us that the statistical rate of q_C^w is atleast τ. □

B. Bounding Box, Counting Oracles and Sampling Oracle

In this section, we provide the proofs of the main theorems required for run-time bound on the max-entropy optimization program. As mentioned earlier, to show that the max-entropy optimization can be performed in polynomial time, we need bounds on the size of the solution of dual program and fast gradient-oracle for the dual function. We first show that size of dual solution can be bounded in terms of the dimension and properties of the dataset. Then we provide a polynomial-time algorithm for first and second-order oracles.

B.1. Bound on size of optimal dual solution

Lemma B.1. *Suppose θ is such that there is an $\eta > 0$ such that, for each $1 \leq i \leq d$, $\eta < \theta_i < 1-\eta$, then the optimal dual solution corresponding to such a θ and q_C^w satisfies*

$$\|\lambda^\star\|_2 \leq \frac{d}{\eta}\log\frac{1}{C}.$$

The proof of this lemma is along similar lines as the proof of bounding box in (Singh & Vishnoi, 2014). The key difference is that the proof in (Singh & Vishnoi, 2014) does not consider a prior on the distribution.

Proof. We are given that θ is in the η-interior of the hypercube, i.e., for each $1 \leq i \leq d$, $\eta < \theta_i < 1-\eta$. Hence a ball of radius η, centered at θ, is contained with the hypercube.

We will first provide a bound for a general prior q and then substitute properties specific to q_C^w. To that end, for a prior q let L_q denote the following quantity,

$$L_q := \log\frac{1}{\min_\alpha q(\alpha)}.$$

To show the bound in Lemma 4.2, we will try to prove that the optimal dual solution, multiplied by a factor of $1/L_q$, lies in a ball of radius $1/\eta$ centered at θ and later provide a bound on L_q. Let

$$\hat{\lambda} = \theta - \frac{\lambda^\star}{L_q}.$$

Firstly, note that we can bound the objective function of (dual-MaxEnt) as follows. Since the objective function of (primal-MaxEnt) is the negative of KL-divergence, it's value is always less than zero. Hence, by strong duality we get

that, for a given prior q,

$$\log\left(\sum_{\alpha\in\{0,1\}^d} q(\alpha)e^{\langle\alpha-\theta,\lambda^\star\rangle}\right)\le 0.$$

This implies that

$$\min_\alpha q(\alpha)\sum_{\alpha\in\{0,1\}^d} e^{\langle\alpha-\theta,\lambda^\star\rangle}\le\sum_{\alpha\in\{0,1\}^d} q(\alpha)e^{\langle\alpha-\theta,\lambda^\star\rangle}\le 1.$$

Therefore, for all $\alpha\in\{0,1\}^d$,

$$e^{\langle\alpha-\theta,\lambda^\star\rangle}\le\frac{1}{\min_\alpha q(\alpha)}.$$

Taking log both sides, we get

$$\langle\alpha-\theta,\lambda^\star\rangle\le\log\frac{1}{\min_\alpha q(\alpha)}=L_q.$$

Substituting $\hat\lambda$, we get

$$\langle\alpha-\theta,\theta-\hat\lambda\rangle\le 1. \tag{1}$$

Note that since this inequality holds for all $\alpha\in\{0,1\}^d$, it also holds for all $\alpha\in\mathrm{conv}\{0,1\}^d$. Next we choose α appropriately so as to bound the distance between θ and $\hat\lambda$. Choose

$$\alpha=\theta+\frac{\theta-\hat\lambda}{\|\theta-\hat\lambda\|}\cdot\eta.$$

Note that $\|\alpha-\theta\|\le\eta$, hence this α lies within the hypercube. Then we can apply (1) to get

$$\left\langle\frac{\theta-\hat\lambda}{\|\theta-\hat\lambda\|}\cdot\eta,\theta-\hat\lambda\right\rangle\le 1.$$

This directly leads to

$$\|\theta-\hat\lambda\|\le\frac{1}{\eta}.$$

Hence we know that $\hat\lambda$ is within a ball of radius $1/\eta$ centered at θ. Substituting the definition of $\hat\lambda$ into this bound, we directly get that

$$\left\|\frac{\lambda^\star}{L_q}\right\|\le\frac{1}{\eta}\implies\|\lambda^\star\|\le\frac{L_q}{\eta}. \tag{2}$$

The above bound is generic for any given prior q. To substitute $q=q_C^w$, we simply need to calculate $L_{q_C^w}$. Note that the prior q_C^w assigns a uniform probability mass to all points not in the dataset $\mathcal{S}$. Hence, for any $\alpha\in\{0,1\}^d$

$$q_C^w(\alpha)\ge\frac{C}{|\Omega|}=\frac{C}{2^d}.$$

Therefore,

$$L_{q_C^w}\le d\log\frac{1}{C}.$$

Substituting the value of $L_{q_C^w}$ in (2), we get

$$\|\lambda^\star\|\le\frac{d}{\eta}\log\frac{1}{C}.$$

□

B.2. Interiority of expected vector

The assumption that θ should be in η-interior the hypercube can translate to an assumption on the "non-redundancy" of the data set, for some natural choice of θ. For example, to maintain consistency with the dataset $\mathcal{S}$, θ can be set to be the following:

$$\theta=\sum_{\alpha\in\mathcal{S}}\frac{n_\alpha}{N}\alpha.$$

This corresponds to the mean of the dataset. In this case, the assumption that for each $1\le i\le d$,

$$\eta<\theta_i$$

implies that more than η-fraction of the elements in the dataset $\mathcal{S}$ have the i-th attribute value 1. Similarly,

$$\theta_i>1-\eta$$

implies that more than η-fraction of the elements in the dataset $\mathcal{S}$ have the i-th attribute value 0.

The reason that this is a non-redundancy assumption is that it implies that no attribute is redundant in the dataset. For example, if for an attribute i, θ_i was 1 it would mean that all elements in $\mathcal{S}$ have the i-th attribute 1 and in that case, we can simply remove the attribute.

B.3. Oracles for dual objective function

Lemma B.2 (Oracles for the dual objective function). *There is an algorithm that, given a reweighted distribution $w:\mathcal{S}\to[0,1]$, and $\theta,\lambda\in\mathbb{R}^d$ computes $h_{\theta,q}(\lambda)$, $\nabla h_{\theta,q}(\lambda)$, and $\nabla^2 h_{\theta,q}(\lambda)$ in time polynomial in N,d and the bit complexities of all the numbers involved: $w(\alpha)$ for $\alpha\in\mathcal{S}$, and e^{λ_i},θ_i for $1\le i\le d$. Here, $q=q_C^w$.*

Proof. For the given prior q and vector θ, let $g_{\theta,q}$ denote the sum, i.e.,

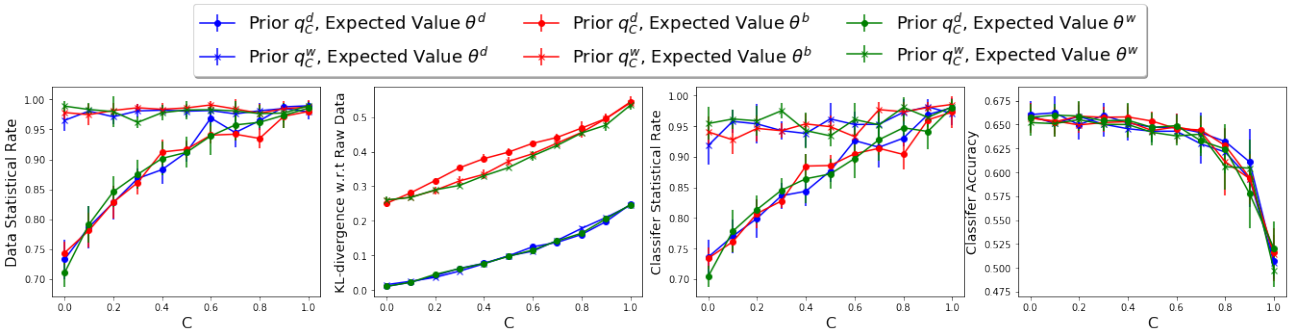
$$g_q(\lambda):=\sum_{\alpha\in\Omega}q(\alpha)e^{\langle\alpha,\lambda\rangle}$$

Then the dual function $h_{\theta,q}(\lambda)$ is

$$h_{\theta,q}(\lambda)=\log\left(g_q(\lambda)\right)-\langle\theta,\lambda\rangle.$$

The main bottleneck in computing the above quantities is evaluating the summation terms. For all three terms, the summation is obtained from the derivative of g_q.

$$\nabla g_q(\lambda)=\sum_{\alpha\in\Omega}\alpha\cdot q(\alpha)e^{\langle\alpha,\lambda\rangle}\text{ and}$$



(a) Data statistical rate vs C (b) KL-divergence w.r.t. raw data vs C (c) Classifier Statistical rate vs C (d) Classifier Accuracy vs C

Figure 1. Comparison of max-entropy distributions with different priors and expectation vectors for small version of COMPAS dataset. Note that a value of C = 1 effectively would result in sampling uniformly at random from the entire domain. Hence, as expected, we see fairness increase and accuracy decrease as C increases. (a) Data statistical rate for COMPAS dataset. We observe that using q_C^{w} is better with respect to statistical rate than using q_C^{d}. The value of C does not significantly affect the results for q_C^{w}; this is expected since q_C^w is constructed to be fair for all C. (b) KL-divergence between the empirical distributions as compared with the raw COMPAS data. We observe that this value is smaller when using the expected vector θ^{d}. (c) Classifier statistical rate vs C. Similar to data statistical rate results for COMPAS dataset, we observe that using the q_C^{w} prior results in a fairer outcome. Here there is a slight increase in fairness as C is increased even for q_C^{w}. (d) Classifier accuracy vs C. We observe that there is no significant difference in accuracy across different metrics and priors. This is surprising, especially in light of the significant differences with respect to how well they capture the raw data.

$$\nabla^2 g_q(\lambda) = \sum_{\alpha \in \Omega} \alpha\alpha^\top \cdot q(\alpha) e^{\langle \alpha, \lambda \rangle}.$$

Then, the gradient and Hessian can be represented using ∇g_q and $\nabla^2 g_q$.

$$\nabla h_{\theta,q}(\lambda) = \frac{1}{g_q(\lambda)} \nabla g_q(\lambda) - \theta,$$

$$\nabla^2 h_{\theta,q}(\lambda) = \frac{1}{g_q(\lambda)} \nabla^2 g_q(\lambda) - \frac{1}{g_q(\lambda)^2} \nabla g_q(\lambda) \nabla g_q(\lambda)^\top.$$

Given the above representation of gradient and oracle, if we are able to compute $g_q(\lambda), \nabla g_q(\lambda), \nabla^2 g_q(\lambda)$ efficiently, then using these to compute $h_{\theta,q}(\lambda)$, $\nabla h_{\theta,q}(\lambda)$ and $\nabla^2 h_{\theta,q}(\lambda)$ just involves constant number of addition and multiplication operations, time taken for which is linear in bit complexities of the numbers involved. Hence we will focus on efficiently evaluating the summations.

Recall that

$$q = q_C^w = C \cdot u + (1 - C) \cdot w.$$

Since $g_q(\lambda), \nabla g_q(\lambda), \nabla^2 g_q(\lambda)$ are all linear in q, we can evaluate the summations separately for u and w.

For w, since the support of the distribution is just the dataset $\mathcal{S}$,

$$g_w(\lambda) = \sum_{\alpha \in \Omega} w(\alpha) e^{\langle \alpha, \lambda \rangle} = \sum_{\alpha \in S} w(\alpha) e^{\langle \alpha, \lambda \rangle}$$

We can directly evaluate the summation using $O(Nd)$ operations (first compute the inner product then summation), where each operation is linear in the bit complexity of w and e^λ. For $\nabla g_w(\lambda)$, we can represent it as

$$g_w(\lambda) = \sum_{\alpha \in S} \alpha \cdot w(\alpha) e^{\langle \alpha, \lambda \rangle}.$$

Once again we can evaluate all inner products using $O(Nd)$ operations and then compute the gradient vector in another $O(Nd)$ operations. In a similar manner, we can also evaluate $\nabla^2 g_w(\lambda)$ in $O(Nd^2)$ operations.

Next we need bounds on the number of operations required for the uniform part of q. The main idea is that if the distribution is uniform over the entire domain, then the summation can be separated in terms of the individual features. For the uniform distribution, let us write λ as $(\lambda_1, \ldots, \lambda_d)$, where λ_i corresponds to ith attribute and let us define variables:

$$\overline{\alpha}_i := \alpha_i \cdot e_i,$$

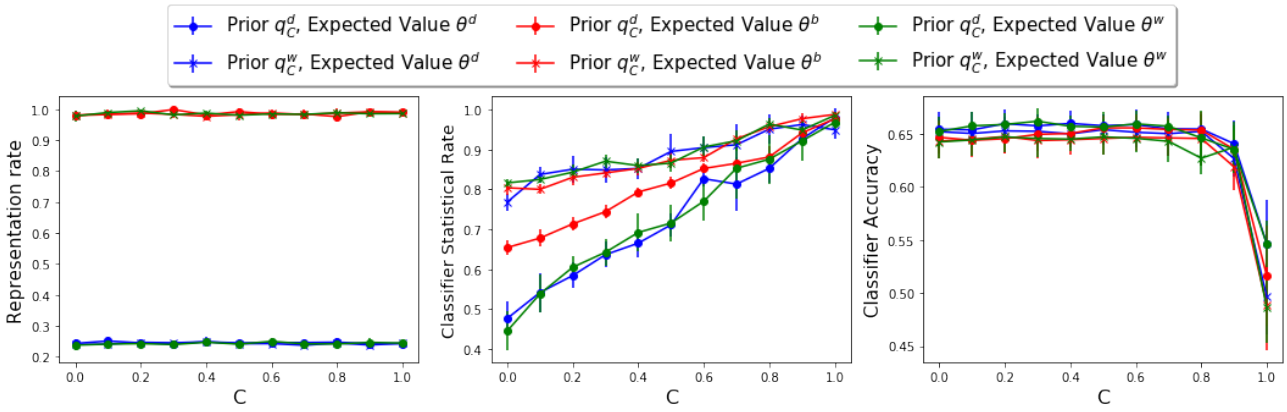
where e_i is the standard basis vector in $\mathbb{R}^d$, with 1 in the i-th location and 0 elsewhere. Let

$$s_i^0 := \sum_{\alpha_i \in \{0,1\}} e^{\lambda_i \cdot \alpha_i},$$

$$s_i^1 := \sum_{\alpha_i \in \{0,1\}} \overline{\alpha}_i e^{\lambda_i \cdot \alpha_i},$$

$$s_i^2 := \sum_{\alpha_i \in \{0,1\}} \overline{\alpha}_i \overline{\alpha}_i^\top e^{\lambda_i \cdot \alpha_i},$$

for all $i \in \{1, \ldots, d\}$ and $\alpha_i \in \{0, 1\}$. Next, we can



(a) Representation rate vs C. (b) Classifier Statistical Rate vs C. (c) Classifier Accuracy vs C.

Figure 2. The figures show the comparison of max-entropy distributions with different prior distributions and expected values. The base dataset is the small version of COMPAS. The first figure show the representation rate of different max-entropy distribution; the representation rate is 1 when using balanced expected vectors, such as θ^w or θ^b. The second and third figure show the statistical rate and accuracy of Gaussian Naive Bayes classifier trained on the output distribution. While the trend across different parameters is the same as observed using decision tree classifier, we note that in this case, the classifier statistical rate is relatively smaller for smaller values of C.

compute the $g_u(\lambda), \nabla g_u(\lambda), \nabla^2 g_u(\lambda)$ using these values.

$$g_u(\lambda) = \frac{1}{|\Omega|}\sum_{\alpha\in\Omega} e^{\langle\alpha,\lambda\rangle} = \frac{1}{|\Omega|}\prod_{i=1}^{d} s_i^0,$$

$$\nabla g_u(\lambda) = \frac{1}{|\Omega|}\sum_{\alpha\in\Omega} \alpha\cdot e^{\langle\alpha,\lambda\rangle} = \frac{1}{|\Omega|}\sum_{i=1}^{d}\left(s_i^1\prod_{j\neq i} s_j^0\right),$$

$$\nabla^2 g_u(\lambda) = \frac{1}{|\Omega|}\sum_{\alpha\in\Omega} \alpha\alpha^\top\cdot e^{\langle\alpha,\lambda\rangle}$$

$$= \frac{1}{|\Omega|}\sum_{i=1}^{d}\left[s_i^2\prod_{j\neq i} s_j^0 + \sum_{j\neq i} s_i^1 (s_j^1)^\top \prod_{k\neq i,j} s_k^0\right].$$

Evaluating $g_u(\lambda)$ involves $(d-1)$ multiplication operations. Similarly, evaluating $\nabla g_u(\lambda)$ involves $O(d^2)$ addition and multiplication operations. Finally, evaluating $\nabla^2 g_u(\lambda)$ involves $O(d^3)$ addition and multiplications operations. Each operation takes time polynomial in the bit complexity of e^λ.

We have shown that for both parts u and w, evaluating the above summations takes time polynomial in the bit complexities of the numbers involved. Since q is a convex combination of u and w, computing $g_u(\lambda)$, $\nabla g_u(\lambda)$ and $\nabla^2 g_u(\lambda)$ also takes time polynomial in the bit complexities of the numbers involved. Specifically, computing $g_u(\lambda)$ requires $O(Nd)$ operations, computing $\nabla g_u(\lambda)$ requires $O(d(N+d))$ operations and computing $g_u(\lambda)$ requires $O(d^2(N+d))$ operations.

□

B.4. Sampling oracle

As stated earlier, the max-entropy distribution $p^\star$ can be succinctly represented using the solution of the dual program $\lambda^\star$. In particular, we have that

$$p^\star(\alpha) = \frac{q(\alpha)e^{\langle\lambda^\star,\alpha\rangle}}{\sum_{\beta\in\Omega} q(\beta)e^{\langle\lambda^\star,\beta\rangle}}.$$

Using the efficient counting oracles of Lemma 4.3 and bounding box of Lemma B.1, we efficiently compute a good approximation to the dual solution $\lambda^\star$. But sampling from the distribution $p^\star$ can still be difficult due to the large domain size. In this section, we show that given $\lambda^\star$ we can efficiently sample from the max-entropy distribution $p^\star$ using the counting oracles described earlier.

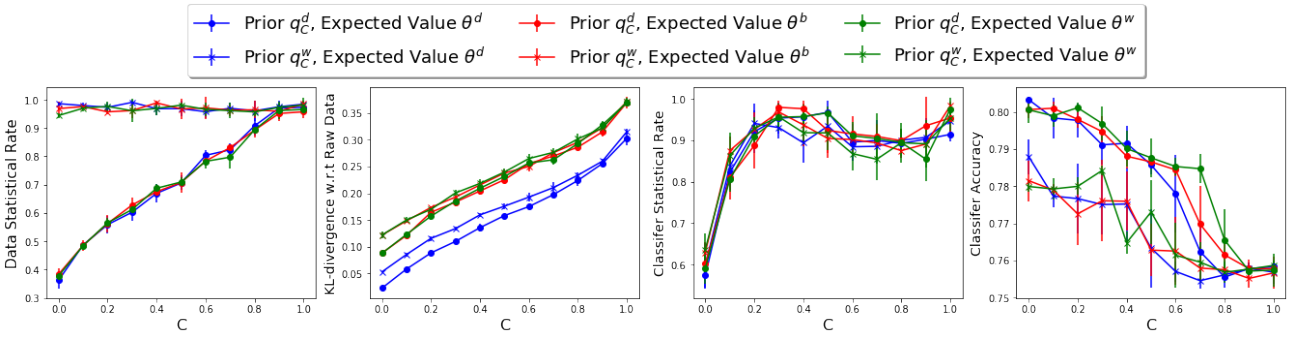
Theorem B.3 (Sampling from counting). *There is an algorithm that, given a weighted distribution $w : \mathcal{S} \to [0,1]$ and $\lambda \in \mathbb{R}^d$, returns a sample from the distribution p, where for any $\alpha \in \Omega$*

$$p(\alpha) = \frac{q_C^w(\alpha)e^{\langle\lambda,\alpha\rangle}}{\sum_{\beta\in\Omega} q_C^w(\beta)e^{\langle\lambda,\beta\rangle}}.$$

The running time of this algorithm is polynomial in N, d and bit complexities of all numbers involved: $w(\alpha)$ for $\alpha \in \mathcal{S}$ and e_i^λ, for $i \in \{1,\ldots,d\}$.

The equivalence of counting and sampling is well-known and a very useful result (Jerrum et al., 1986). We provide the proof for our setting here, for the sake of completion.

Proof. As mentioned before, the goal is to sample from the



(a) Data statistical rate vs C (b) KL-divergence w.r.t. raw data vs C (c) Classifier Statistical rate vs C (d) Classifier Accuracy vs C

Figure 3. Comparison of max-entropy distributions with different priors and expectation vectors for small version of Adult dataset. (a) Data statistical rate for Adult dataset. Once again using q_C^{w} is better with respect to statistical rate than using q_C^{d}. (b) KL-divergence between the empirical distributions as compared with the raw Adult data. We observe that this value is smaller when using the expected vector θ^{d}. However, in this case the gap between divergence when using q_C^w and divergence when using q_C^d is smaller than observed with COMPAS. (c) Classifier statistical rate vs C. In this case, using even q_C^d achieves relatively good statistical rate. However, the statistical rate of max-entropy distributions using q_C^w is slightly better in most cases. (d) Classifier accuracy vs C. As expected, classifier accuracy is higher for distributions using q_C^d than distributions using q_C^w. This is because q_C^w involves weighing the samples in a manner that is not always consistent with the frequency of the samples.

distribution

$$p(\alpha) = \frac{q_C^w(\alpha)e^{\langle\lambda,\alpha\rangle}}{\sum_{\beta\in\Omega} q_C^w(\beta)e^{\langle\lambda,\beta\rangle}}.$$

The primary bottleneck in sampling is evaluating the normalizing term,

$$\sum_{\beta\in\Omega} q_C^w(\beta)e^{\langle\lambda,\beta\rangle}.$$

To evaluate this sum, we have an efficient oracle, i.e, the counting oracle from Lemma 4.3. The lemma (and the algorithm) allow us to calculate the sum in $O(Nd)$ operations, where each operation has bit complexity polynomial in the numbers involved: $w(\alpha)$ for $\alpha \in \Omega$ and e^{λ}. Hence, we can evaluate the normalizing term efficiently.

However, we still cannot sample by enumerating all probabilities since the size of the domain is exponential. To efficiently sample from the distribution, we sample each feature of α individually. Let A denote the random variable with probability distribution p. Let A_1 denote the element at the first position of A.

$$\begin{aligned}\mathbb{P}[A_1 = 0] &= \frac{\sum_{\alpha\in\Omega|\alpha_1=0} q_C^w(\alpha)e^{\langle\lambda,\alpha\rangle}}{\sum_{\beta\in\Omega} q_C^w(\beta)e^{\langle\lambda,\beta\rangle}} \\ &= \frac{\sum_{\hat{\alpha}\in\Omega^{(1)}} q_{C,1}^w(\hat{\alpha})e^{\langle\lambda^{(1)},\hat{\alpha}\rangle}}{\sum_{\beta\in\Omega} q_C^w(\beta)e^{\langle\lambda,\beta\rangle}}.\end{aligned}$$

Here $\lambda^{(1)}$ is λ without the first element, $\Omega^{(1)}$ is the subdomain of all feature except the first feature and $q_{C,1}^w$ is the distribution q_C^w conditional on the first feature being always 0. Note that $q_{C,1}^w$ is a distribution supported on $\Omega^{(1)}$, and we can use the counting oracle of Lemma 4.3 to calculate the sum

$$\sum_{\hat{\alpha}\in\Omega^{(1)}} q_{C,1}^w(\hat{\alpha})e^{\langle\lambda^{(1)},\hat{\alpha}\rangle}$$

in $O(N(d-1))$ operations. Hence we can calculate the probability $\mathbb{P}[A_1 = 0]$ in $O(Nd)$ operations. Then we can do a coin toss, whose tail probability is chosen to be $\mathbb{P}[A_1 = 0]$, and set $\alpha_1 = 1$ if we heads and $\alpha_1 = 0$ otherwise. Next depending on the value we get for α_1, we can calculate the marginal probability of α_2 being 0. Say $\alpha_1 = a_1$. Then

$$\mathbb{P}[A_1 = 0] = \frac{\sum_{\alpha\in\Omega|\alpha_1=a_1,\alpha_2=0} q_C^w(\alpha)e^{\langle\lambda,\alpha\rangle}}{\sum_{\beta\in\Omega|\beta_1=a_1} q_C^w(\beta)e^{\langle\lambda,\beta\rangle}}.$$

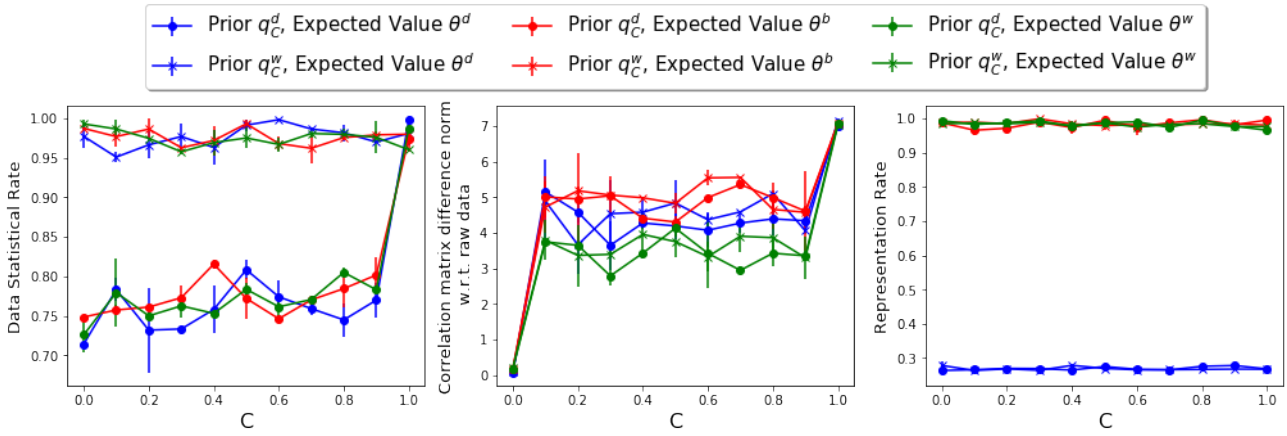
We can repeat the above process of calculating these summations using the counting oracle and once again sample a value of α_2 using the biased coin toss. Repeating this process d times, we get a sample from the distribution p. The number of operations required is $O(Nd^2)$, where each operation has bit complexity polynomial in the numbers involved: $w(\alpha)$ for $\alpha \in \Omega$ and e^{λ}.

□

C. Fairness guarantees

In this section, we provide the proof of the statistical rate bound (Theorem 4.5).

Theorem C.1 (Fairness guarantees). *Given the dataset*



(a) Data statistical rate vs C (b) Utility vs C (c) Representation rate vs C

Figure 4. Comparison of statistical rate, representation rate and correlation matrix difference with respect to raw data for max-entropy distributions with different priors and expected values. The base dataset is the large version of COMPAS.

$\mathcal{S}$, protected attribute ℓ and parameters $\tau, C \in [0,1]$, let $w : S \to [0,1]$ be the reweighted distribution obtained from Algorithm 1. Suppose θ is a vector that satisfies

$$\frac{1}{2} \leq \theta_\ell \leq \frac{1}{1+\tau}.$$

The max-entropy distribution $p^\star$ corresponding to the prior distribution q_C^w and expected value θ has statistical rate atleast τ', where

$$\tau' = \tau - \frac{4\delta \cdot (1+\tau)}{C + 4\delta}.$$

and $\delta = \max_{z\in\{0,1\}} |p^\star(Y = y, Z = z) - q_C^w(Y = y, Z = z)|$; here Y is the random variable when the distribution is restricted to $\mathcal{Y}$ and Z is the random variable when the distribution is restricted to Ω_ℓ.

Proof. The proof of this theorem uses the bounds on the distribution of q_C^w that are obtained from Lemma A.4. By the definition of δ, we have that

$$\begin{aligned} q_C^w(Y = y, Z = z) - \delta &\leq p^\star(Y = y, Z = z) \\ &\leq q_C^w(Y = y, Z = z) + \delta. \end{aligned}$$

Using this inequality, we can bound the ratio of the above term for different sensitive attributes as

$$\frac{p^\star(Z = z_1, Y = y)}{p^\star(Z = z_2, Y = y)} \geq \frac{q_C^w(Y = y, Z = z_1) - \delta}{q_C^w(Y = y, Z = z_2) + \delta}.$$

Next, applying Lemma A.4, with $v_1 = u$ and $v_2 = v^w$, we have the following properties of q_C^w

$$q_C^w(Y = y, Z = 0) \geq \tau \cdot q_C^w(Y = y, Z = 1),$$

and

$$q_C^w(Y = y, Z = 1) \geq q_C^w(Y = y, Z = 0).$$

Furthermore, since q_C^w assigns a uniform mass to all points in Ω, we can also get a lower bound on $q_C^w(Y = y, Z = z_2)$.

$$\begin{aligned} q_C^w(Y = y, Z = z) &= \sum_{\alpha | y(\alpha)=y, z(\alpha)=z} q_C^w(\alpha) \\ &\geq \sum_{\alpha | y(\alpha)=y, z(\alpha)=z} \frac{C}{|\Omega|} = \frac{C}{2|\mathcal{Y}|}. \end{aligned}$$

We can now use the fairness guarantee on q_C^w and lower bound for distribution to get the ratio bounds for max-entropy distribution.

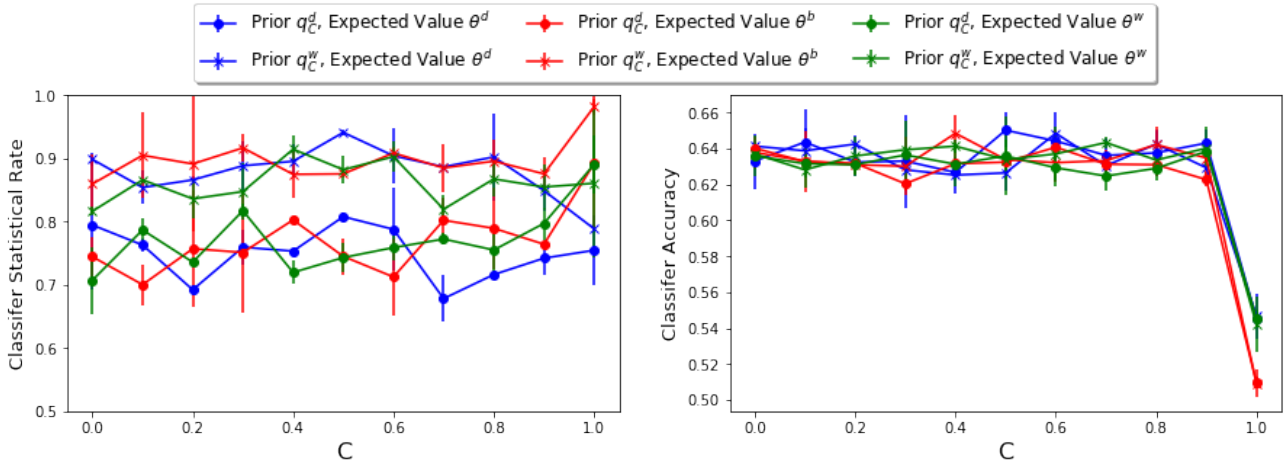
$$\begin{aligned} \frac{p^\star(Y = y, Z = 0)}{p^\star(Y = y, Z = 1)} &\geq \frac{\tau \cdot q_C^w(Y = y, Z = 1) - \delta}{q_C^w(Y = y, Z = 1) + \delta} \\ &= \tau - \delta \cdot \frac{(1+\tau)}{q_C^w(Y = y, Z = 1) + \delta} \\ &\geq \tau - \delta \cdot \frac{(1+\tau)}{\frac{C}{2|\mathcal{Y}|} + \delta}. \end{aligned}$$

By the choice of θ, we know that

$$1 - \theta_\ell > \theta_\ell \implies p^\star(Z = 1) \geq p^\star(Z = 0).$$

Therefore,

$$\begin{aligned} \frac{p^\star(Y = y \mid Z = 0)}{p^\star(Y = y \mid Z = 1)} &= \frac{p^\star(Y = y, Z = 0)}{p^\star(Y = y, Z = 1)} \cdot \frac{p^\star(Z = 1)}{p^\star(Z = 0)} \\ &\geq \tau - \delta \cdot \frac{(1+\tau)}{\frac{C}{2|\mathcal{Y}|} + \delta}. \end{aligned}$$



(a) Classifier statistical rate vs C

(b) Classifier accuracy vs C

Figure 5. Comparison of Decision Tree classifier trained on data from different max-entropy distributions with different prior distributions and expected values. The base dataset is the large version of COMPAS.

Similarly, for the other direction of this ratio, we can get

$$\begin{aligned}\frac{p^\star(Y=y,Z=1)}{p^\star(Y=y,Z=0)} &\geq \frac{q_C^w(Y=y,Z=0)-\delta}{q_C^w(Y=y,Z=0)+\delta}\\ &= 1-\delta\cdot\frac{2}{q_C^w(Y=y,Z=0)+\delta}\\ &\geq 1-\delta\cdot\frac{2}{\frac{C}{2|\mathcal{Y}|}+\delta}.\end{aligned}$$

Once again,

$$1-\theta_\ell > \tau\cdot\theta_\ell \implies p^\star(Z=0)\geq p^\star(Z=1).$$

Therefore,

$$\begin{aligned}\frac{p^\star(Y=y\mid Z=1)}{p^\star(Y=y\mid Z=0)} &= \frac{p^\star(Y=y,Z=1)}{p^\star(Y=y,Z=0)}\cdot\frac{p^\star(Z=0)}{p^\star(Z=1)}\\ &\geq \tau\left(1-\delta\cdot\frac{2}{\frac{C}{2|\mathcal{Y}|}+\delta}\right).\end{aligned}$$

Note that

$$\tau\left(1-\delta\cdot\frac{2}{\frac{C}{2|\mathcal{Y}|}+\delta}\right)\geq\tau-\delta\cdot\frac{(1+\tau)}{\frac{C}{2|\mathcal{Y}|}+\delta}.$$

Using $|\mathcal{Y}|=2$, we get that the statistical rate is atleast

$$\tau-\frac{4\delta\cdot(1+\tau)}{C+4\delta}.$$

□

D. Additional details and empirical results for small COMPAS and Adult datasets

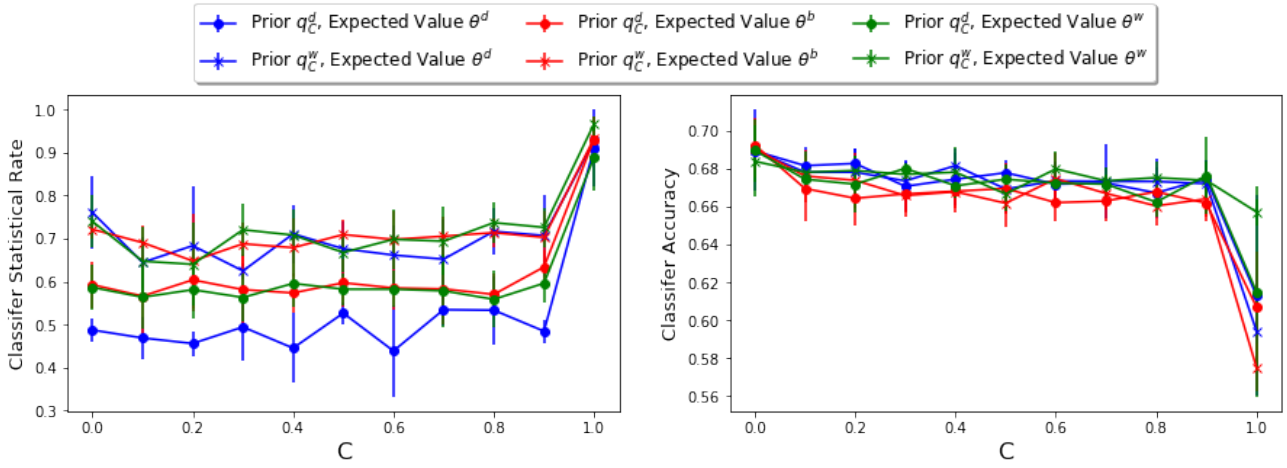
Features of Adult dataset. The demographic features used from this dataset are gender, race, age and years of education. The age attribute in this case is categorized by decade, with 7 categories (the last one being age $\geq$ 70 years). The education years attribute is also a categorical attribute, with the categories being $(<6), 6, 7, \cdots, 12, (>12)$ years. The label is a binary marker indicating whether the annual income is greater than \$50K or not.

Features of small version of COMPAS dataset. For this dataset, we use the features gender, race, age, priors count, and charge degree as features, and a binary marker of recidivism within two years as the label.

D.1. Graphical representation of results in Table 2

We present all the results in Table 2 in a graphical form in Figure 7. This figure also includes the results for COMPAS dataset using race as the protected attribute (encoded as binary, i.e., "Caucasian vs Non-Caucasian").

The plots show that, in case of race of as the protected attribute, once again the statistical rate and representation rate of max-entropy distribution is close to 1 and much better than the raw dataset. The distance of max-entropy distribution from the empirical distribution of the raw dataset is smaller in this case, since the correction required to enforce fairness is smaller than the case when gender is the protected attribute.



(a) Classifier statistical rate vs C

(b) Classifier accuracy vs C

Figure 6. Comparison of Gaussian Naive Bayes classifier trained on data from max-entropy distributions with different prior distributions and expected values. The base dataset is the large version of COMPAS.

D.2. Empirical results using different priors, marginal vectors and smoothing parameters.

Given training data $\mathcal{S}$, we can estimate different maximum entropy distributions with given parameters using $\mathcal{S}$. We use two kinds of prior distributions: (1) q_C^{d} assigns uniform weights to the samples, i.e.,

$$w = \{n_\alpha / N\}_{\alpha \in \mathcal{S}}.$$

(2) q_C^{w} assigns weights returned by the Algorithm 1 (also used for results in Table 2).

We use three kinds of expectation vectors: (a) the expected value of the dataset $\mathcal{S}$,

$$\theta^d := \left(\sum_{\alpha \in \mathcal{S}} \frac{n_\alpha}{N} X_\alpha, \sum_{\alpha \in \mathcal{S}} \frac{n_\alpha}{N} Y_\alpha, \sum_{\alpha \in \mathcal{S}} \frac{n_\alpha}{N} Z_\alpha \right).$$

The resulting max-entropy distribution is our best guess for the underlying distribution without any modification for fairness. (b) θ^b and (c) θ^w, as defined in Remark 4.6.

This results in six distributions; we generate a synthetic datasets from each distribution to use in our evaluation. We compare the statistical rate, representation rate, divergence from empirical distribution and classifier performance of datasets from these distributions, for varying values of parameter C.

D.2.1. Comparison across priors and expected value vectors

We first evaluate the dataset generated using max-entropy distributions with different combinations of prior weights and expected value mentioned earlier. The results for this evaluation are present in Figure 1 and Figure 3.

Figure 1a and Figure 3a show that for both COMPAS and Adult datasets, the max-entropy distributions obtained using prior q_C^{w} achieve higher statistical rate than the distributions obtained using q_C^{d}. However, the KL-divergence of the max-entropy distributions obtained using expected value θ^{w} or θ^{b} are higher as well. As the samples in the raw dataset are unbalanced with respect to gender, the distributions using balanced marginal distributions (i.e., q_C^w) are expected to have a larger divergence from the empirical distribution of raw data than the distributions using the expected value of data.

Note that, according to the application, one can aim to achieve high representation rate or high statistical rate or both in the final distribution. The max-entropy distribution using q_C^{w} and θ^{d} achieves high statistical rate and low representation rate, while the max-entropy distribution using q_C^{w} and θ^{b} achieves high statistical rate and high representation rate.

D.2.2. Comparison of classifier trained using different max-entropy distribution datasets

For the decision tree classifier trained on the generated data, we compute the statistical rate using the predictions to evaluate the effects of different training data on the fairness of the classifier. In addition, we report the classifier accuracy when trained on each output dataset. The classifier results are presented in Figure 1c,d and Figure 3c,d.

Once again the the max-entropy distributions obtained using

prior distribution q_C^{w} achieve better classifier statistical rate than the distributions obtained using q_C^{d}. The accuracy of the classifiers trained on datasets obtained using prior distribution q_C^{w} is slightly lower than the accuracy of the classifiers trained on distributions obtained using sample uniform weights. However, it is interesting to note that the significant difference in "accuracy" of the data all but disappears when passed through the classifier. Importantly, the accuracy drops sharply as the value of C increases as $C = 1$ assigns equal probability mass to all points in the domain and ignores the original samples. This suggests a C value in the low-to-mid range would likely optimize accuracy and statistical rate simultaneously.

Figure 2b,c presents the Gaussian Naive Bayes classifier statistical rate and accuracy, when trained using different max-entropy distributions on the COMPAS dataset.

D.2.3. Comparison of representation rate

Figure 2a shows the variation of representation rate. As expected, distributions obtained using expected value θ^{b} or θ^{w} have representation rate close to 1.

E. Additional empirical results on larger COMPAS dataset

In this section, we present additional empirical results on the larger version of the COMPAS dataset. In the small version of the dataset, the features used were sex, race, age, priors count, and charge degree as features, and uses a binary marker of recidivism within two years as the label. The age attribute was categorized into three categories, younger than 25, between 25 and 45, and older than 45, and the priors count attribute is categorized in to three categories (no prior crime, between 1 and 3, and more than 3). Further, we only considered data for convicted criminals labelled as being either White or Black.

The large dataset consists of attributes sex, race, age, juvenile felony count, juvenile misdemeanor count, juvenile other count, months in jail, priors count, decile score, charge degree, violent crime, violent recidivism, drug related crime, firearm involved, minor involved, road safety hazard, sex offense, fraud and petty crime, with recidivism as the label. We did not exclude any samples and we did not categorize any attributes. The original data contains samples from 6 different races whose age ranged from 18 to 96 with at most 40 prior counts, juvenile felony count, juvenile misdemeanor count, and juvenile other count.

We model the domain Ω_L for this version as $\{0,1\}^8 \times \{0,1,2\}^3 \times \{0,1,\dots 5\} \times \Delta_6 \times \{0,1,\dots 7\}^2 \times \{0,1,\dots 10\}^2 \times \{0,1,\dots 11\} \times \{0,1,\dots 13\}$. Overall the domain contains approximately 1.4×10^{11} different points.

E.1. Evaluating the statistical rate and accuracy of generated dataset

We evaluate the dataset generated using different max-entropy algorithms. We run the algorithm with different combinations of prior weights and expected value mentioned earlier. We vary the C value for our framework and measure the statistical rate of the output distribution.

For this dataset, calculating the KL-divergence from empirical distribution is difficult due to the large domain size. Hence we consider another metric to check how well the max-entropy distribution preserves the pairwise correlation between features. To calculate this, we first calculate the covariance matrix of the output dataset, say $\mathrm{Cov}_{\mathrm{output}}$ and the original raw dataset $\mathrm{Cov}_{\mathrm{data}}$, and then report the Frobenius norm of the difference of these matrices, i.e., $\|\mathrm{Cov}_{\mathrm{output}} - \mathrm{Cov}_{\mathrm{data}}\|_F^2$. The lower the value of the norm, the better the output distribution preserves the pairwise correlation. The results for this evaluation are present in Figure 4. Here again the first part of the figure shows that the max-entropy distributions obtained using prior q_C^{w} and expected value θ^{w} or θ^{b} achieve higher statistical rate values than the distributions obtained from max-entropy distribution obtained using uniform weights on samples. Similarly the representation rate of max-entropy distributions using prior distribution q_C^{w} and expected value θ^{w} or θ^{b} are close to 1.0.

E.2. Evaluating the statistical rate and accuracy of classifier trained on generated dataset

As mentioned earlier, we use the generated datasets to train a Gaussian Naive Bayes and the Decision Tree Classifier and evaluate the fairness and the accuracy of the resulting classifier.

Firstly, we again vary the C value for our framework and measure the statistical rate of the output of the classifier as well as the accuracy. The results for this evaluation using Gaussian Naive Bayes are present in Figure 6 and using Decision Tree Classifier are present in Figure 5. As expected, once again the the max-entropy distributions obtained using prior distribution q_C^{w} achieve higher statistical rate values than the distributions obtained from max-entropy distribution obtained using uniform weights on samples. The accuracy also drops as the value of C tends to 1. This is again because the prior distribution in case of $C = 1$ assigns equal probability mass to all points in the domain.

F. Full algorithm for max-entropy optimization

In this section, we state the full-algorithm for max-entropy optimization. The algorithm is based on the second-order framework of (Allen Zhu et al., 2017; Cohen et al., 2017).

Algorithm 1 Value-Oracle: Computing dual function value at any point λ

Input: samples $\mathcal{S} := \{\alpha_i\}_{i\in N} \subseteq \{0,1\}^n$, weights $w \in \Delta_{N-1}$, smoothing parameter $C \in [0,1]$ expected vector θ and vector λ
$g_1 \leftarrow 1$
for $j \in \{1,\ldots,n\}$ **do**
  $s_j^0 \leftarrow e^{\lambda_j}/2$
  $g_1 \leftarrow g_1 \cdot s_j^0$
end for
$g_2 \leftarrow 0$
for $i \in \{1,\ldots,N\}$ **do**
  $g_2 \leftarrow g_2 + w_i \cdot e^{\langle \alpha_i, \lambda\rangle}$
end for
$g \leftarrow Cg_1 + (1-C)g_2$
return $\log(g) - \langle \theta, \lambda\rangle$

We start with a complete algorithm for value, gradient and Hessian oracles for h_{θ,q_C^w}, constructed along similar lines as the proof of Lemma 4.3.

F.1. Oracle algorithm

Algorithm 1 shows how to compute the dual function h_{θ,q_C^w} value at any point λ, Algorithm 2 shows how to compute the gradient of the dual function at any point λ, and Algorithm 2 shows how to compute the Hessian of the dual function at any point λ.

F.2. Max-entropy optimization algorithm

With the first and second order oracles, we can now state our entire algorithm for the hypercube domain. Algorithm 4 presents the approach to optimizing the dual of the max-entropy program. The inner optimization problem (inner-Opt) is a normal quadratic optimization problem and can be solved in polynomial time using standard interior-point methods (Karmarkar, 1984; Wright, 2005).

F.3. Time complexity of Algorithm 4

To provide a time complexity bound for Algorithm 4, we will invoke the bounds proved by (Cohen et al., 2017) for optimization of *second-order robust* functions. In particular, the algorithm runs in time polynomial in d, N and the bit complexity of the number, provided

1. there is a bound on the size of dual solution, $\lambda^\star$,
2. efficient first and second-order oracles for the dual function,
3. the dual function is *second-order robust*.

We have already shown that $\|\lambda^\star\|$ is bounded (Lemma 4.2) as well as provided fast first and second-order oracles (Lemma 4.3). To establish to polynomial time complexity

Algorithm 2 Gradient-Oracle: Computing gradient of dual function at any point λ

Input: samples $\mathcal{S} := \{\alpha_i\}_{i\in N} \subseteq \{0,1\}^n$, weights $w \in \Delta_{N-1}$, smoothing parameter $C \in [0,1]$ expected vector θ and vector λ
$g_1 \leftarrow 0$
for $j \in \{1,\ldots,n\}$ **do**
  $s_j^0 \leftarrow e^{\lambda_j}/2$
  $s_j^1 \leftarrow e_j \cdot e^{\lambda_j}/2$ {e_j is standard basis vector with 1 in j-th location}
end for
for $j \in \{1,\ldots,n\}$ **do**
  $t \leftarrow 1$
  for $k \in \{1,\ldots,n\} \setminus \{j\}$ **do**
    $t \leftarrow t \cdot s_k^0$
  end for
  $g_1 \leftarrow g_1 + s_j^1 \cdot t$
end for
$g_2 \leftarrow 0$
for $i \in \{1,\ldots,N\}$ **do**
  $g_2 \leftarrow g_2 + \alpha \cdot w_i \cdot e^{\langle \alpha_i, \lambda\rangle}$
end for
$g \leftarrow Cg_1 + (1-C)g_2$
$v \leftarrow$Value-Oracle $(S, w, C, \theta, \lambda) + \langle \theta, \lambda\rangle$
$v_2 \leftarrow e^v$
return $\frac{1}{v_2} g - \theta$

of this algorithm, we just need to prove that dual function is second-order robust. A convex function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be α-second order robust, if for all $x, y \in \mathbb{R}^n$ with $\|y\|_\infty \le 1$ satisfies

$$\left|D^3 f(x)[y,y,y]\right| \le \alpha D^2 f(x)[y,y]$$

where $D^k f(x)[y,\ldots,y] := \frac{d^k}{dt^k} f(x+ty)\Big|_{t=0}$. The following lemma establishes the second-order robustness of the dual function $h_{\theta,q}$.

Lemma F.1 (Second-order robustness of the dual–MaxEnt function). *Given $\Omega = \{0,1\}^n$, prior $q : \Omega \to [0,1]$ and the target expected vector $\theta \in \mathrm{conv}(\Omega)$, the dual maximum entropy function $h_{\theta,q}(\lambda) := \log\left(\sum_{\alpha\in\Omega} q(\alpha) e^{\langle \lambda, \alpha-\theta\rangle}\right)$ is $4n$-second order robust.*

Using this second-order robustness property, bound on $\|\lambda^\star\|$, gradient, Hessian oracles and interior point method to solve the inner-optimization problem (inner-Opt), as a corollary of Theorem 3.4 in (Cohen et al., 2017), it follows that Algorithm 4 runs in time polynomial in d, N and bit complexities of all the numbers involved.

Before proving the lemma, we state and prove the following general claim in the proof.

Claim F.2. *Let X be a real valued random variable over*

Algorithm 3 Hessian-Oracle: Computing hessian of dual function at any point λ

Input: samples $\mathcal{S} := \{\alpha_i\}_{i\in N} \subseteq \{0,1\}^n$, weights $w \in \Delta_{N-1}$, smoothing parameter $C \in [0,1]$ expected vector θ and vector λ
$g_1 \leftarrow 0$
for $j \in \{1,\dots,n\}$ **do**
  $s_j^0 \leftarrow (e^{(1-\theta_j)\lambda_j} + e^{-\theta_j\lambda_j})/2$
  $s_j^1 \leftarrow e_j \cdot (e^{(1-\theta_j)\lambda_j})/2$ {e_j is standard basis vector with 1 in j-th location}
  $s_j^2 \leftarrow e_j e_j^\top \cdot (e^{(1-\theta_j)\lambda_j})/2$
end for
for $j \in \{1,\dots,n\}$ **do**
  $t_1 \leftarrow 1$
  $t_2 \leftarrow 0$
  for $k \in \{1,\dots,n\} \setminus \{j\}$ **do**
    $t_1 \leftarrow t_1 \cdot s_k^0$
    $t_3 \leftarrow 1$
    for $l \in \{1,\dots,n\} \setminus \{j,k\}$ **do**
      $t_3 \leftarrow t_3 \cdot s_l^0$
    end for
    $t_2 \leftarrow t_2 + s_j^1 {s_k^1}^\top \cdot t_3$
  end for
  $g_1 \leftarrow g_1 + s_i^2 \cdot t_1 + t_2$
end for
$g_2 \leftarrow 0$
for $i \in \{1,\dots,N\}$ **do**
  $g_2 \leftarrow g_2 + \alpha\alpha^\top \cdot w_i \cdot e^{\langle \alpha_i - \theta, \lambda\rangle}$
end for
$g \leftarrow Cg_1 + (1-C)g_2$
$v_1 \leftarrow$Value-Oracle $(S, w, C, \theta, \lambda)$
$v_2 \leftarrow$Gradient-Oracle $(S, w, C, \theta, \lambda)$
$v_3 \leftarrow \frac{1}{v_1} g - (v_2+\theta)(v_2-\theta)^\top$
return v_3

Algorithm 4 Full algorithm to compute max-entropy distributions

Input: samples $\mathcal{S} := \{(X_i, Y_i, Z_i)\}_{i\in N} \subseteq \{0,1\}^n$, parameter $C \in [0,1]$, target expected value θ, weights $\{w_i\}_{i=1}^N \in \Delta_{N-1}$ and $\varepsilon > 0$
$q_C^w \leftarrow$ Prior distribution constructed using $\{w_i\}_{i=1}^N$ and C
$R \leftarrow 8n \log 1/C\varepsilon$
$T \leftarrow 16nR \log 1/C\varepsilon$
$\lambda \leftarrow \mathbf{0}$
for $i = 1$ **to** T **do**
  $g \leftarrow$ Gradient-Oracle $(S, w, C, \theta, \lambda)$
  $H \leftarrow$ Hessian-Oracle $(S, w, C, \theta, \lambda)$
  $y_\varepsilon \leftarrow \frac{\varepsilon}{8nR}$-approximate minimizer of the following convex quadratic program (using primal path following algorithm (Karmarkar, 1984; Wright, 2005)),

$$\inf_{y\in\mathbb{R}^n} \langle g, y\rangle + \frac{1}{2e} y^\top H y$$
$$\text{s.t. } \|y\|_\infty \le \frac{1}{8n} \quad \text{and } \|\lambda + y\|_\infty \le R \qquad \text{(inner-Opt)}$$

  $\lambda \leftarrow \lambda + y_\varepsilon/e^2$
end for
return λ

the discrete set Ω with $|X| \le r$ for some constant $r \in \mathbb{R}_+$. Then,

$$|\mathbb{E}\left[X^3\right] - \mathbb{E}\left[X^2\right]\mathbb{E}\left[X\right]| \le 2r(\mathbb{E}\left[X^2\right] - \mathbb{E}\left[X\right]^2).$$

Proof. Let us denote the probability mass function of X with p. Then,

$$\begin{aligned}
&\mathbb{E}\left[X^3\right] - \mathbb{E}\left[X^2\right]\mathbb{E}\left[X\right] \\
&\quad= \sum_{\alpha\in\Omega} X(\alpha)^3 p(\alpha) - \sum_{\alpha,\beta\in\Omega} X(\alpha)^2 X(\beta) p(\alpha)p(\beta) \\
&\quad= \frac{1}{2}\sum_{\alpha,\beta\in\Omega} (X(\alpha)^3 - X(\alpha)^2X(\beta))p(\alpha)p(\beta) \\
&\qquad+ \frac{1}{2}\sum_{\alpha,\beta\in\Omega} (X(\beta)^3 - X(\alpha)X(\beta)^2)p(\alpha)p(\beta) \\
&\quad= \frac{1}{2}\sum_{\alpha,\beta\in\Omega} (X(\alpha) - X(\beta))^2(X(\alpha)+X(\beta))p(\alpha)p(\beta).
\end{aligned}$$

We also note that, $|X(\alpha) + X(\beta)| \le 2r$ for any $\alpha, \beta \in \Omega$ as $|X| \le r$. Therefore,

$$\begin{aligned}
&|\mathbb{E}\left[X^3\right] - \mathbb{E}\left[X^2\right]\mathbb{E}\left[X\right]| \\
&\quad= \frac{1}{2}\left|\sum_{\alpha,\beta\in\Omega} (X(\alpha) - X(\beta))^2(X(\alpha)+X(\beta))p(\alpha)p(\beta)\right| \\
&\quad\le r\sum_{\alpha,\beta\in\Omega} (X(\alpha) - X(\beta))^2 p(\alpha)p(\beta) \\
&\quad= 2r(\mathbb{E}\left[X^2\right] - \mathbb{E}\left[X\right]^2).
\end{aligned}$$

□

Proof of Lemma F.1. Let us fix a point $\lambda_0 \in \mathbb{R}^n$ and a direction $\lambda_1 \in \mathbb{R}^n$ with $\|\lambda_1\|_\infty \le 1$. We need to verify that

$$\left|D^3 h_{\theta,q}(\lambda_0)[\lambda_1,\lambda_1,\lambda_1]\right| \le 4nD^2 h_{\theta,q}(\lambda_0)[\lambda_1,\lambda_1] \quad (2)$$

to show that $h_{\theta,q}$ is $4n$-second order robust.

For any $k \in \mathbb{Z}$, let $g_q^{(k)}$ denote the following function.

$$g_q^{(k)}(\lambda_0, \lambda_1) = \sum_{\alpha \in \Omega} q(\alpha) \cdot \langle \lambda_1, \alpha \rangle^k \cdot e^{\langle \lambda_0, \alpha \rangle},$$

Then the derivative $D^2 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1]$ can be written as

$$D^2 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1] = \frac{g_q^{(2)}(\lambda_0, \lambda_1)}{g_q^{(0)}(\lambda_0, \lambda_1)} - \frac{g_q^{(1)}(\lambda_0, \lambda_1)^2}{g_q^{(0)}(\lambda_0, \lambda_1)^2}$$

Similarly,

$$D^3 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1, \lambda_1] = \frac{g_q^{(3)}(\lambda_0, \lambda_1)}{g_q^{(0)}(\lambda_0, \lambda_1)} + \frac{2g_q^{(1)}(\lambda_0, \lambda_1)^3}{g_q^{(0)}(\lambda_0, \lambda_1)^3} - \frac{3g_q^{(2)}(\lambda_0, \lambda_1) g_q^{(1)}(\lambda_0, \lambda_1)}{g_q^{(0)}(\lambda_0, \lambda_1)^2}$$

We begin by dividing $D^3 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1, \lambda_1]$ into two parts, and prove upper bounds on each part individually. Firstly note that using Cauchy-Swartz, we can bound $g_q^{(1)}$ using $g_q^{(0)}$ in the following way,

$$\begin{aligned} g_q^{(1)}(\lambda_0, \lambda_1) &= \sum_{\alpha \in \Omega} q(\alpha) \cdot \langle \lambda_1, \alpha \rangle \cdot e^{\langle \lambda_0, \alpha \rangle} \\ &\leq \sum_{\alpha \in \Omega} q(\alpha) \cdot \|\lambda_1\|_\infty \|\alpha\|_1 \cdot e^{\langle \lambda_0, \alpha \rangle} \\ &\leq \max_{\alpha \in \Omega} \|\alpha\|_1 \cdot g_q^{(0)}(\lambda_0, \lambda_1) \\ &\leq n \cdot g_q^{(0)}(\lambda_0, \lambda_1) \end{aligned}$$

since $\|\lambda_1\|_\infty \leq 1$ and $\max_{\alpha \in \Omega} \|\alpha\|_1 \leq n$, as all features in Ω are binary. Now using this property, we get that

$$\begin{aligned} &\left| \frac{2g_q^{(1)}(\lambda_0, \lambda_1)^3}{g_q^{(0)}(\lambda_0, \lambda_1)^3} - \frac{2g_q^{(2)}(\lambda_0, \lambda_1) g_q^{(1)}(\lambda_0, \lambda_1)}{g_q^{(0)}(\lambda_0, \lambda_1)^2} \right| \\ &= \left| \frac{2g_q^{(1)}(\lambda_0, \lambda_1)}{g_q^{(0)}(\lambda_0, \lambda_1)} \right| \cdot D^2 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1] \\ &\leq 2n \cdot D^2 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1]. \end{aligned} \tag{3}$$

Next we try to bound the second part of $D^3 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1, \lambda_1]$. To do so, let $p_{\lambda_0} : \Omega \to [0, 1]$ denote the following distribution

$$p_{\lambda_0}(\alpha) = \frac{q(\alpha) e^{\langle \lambda_0, \alpha \rangle}}{g_q^{(0)}(\lambda_0, \lambda_1)}.$$

Then using Claim F.2 and the fact $\max_{\alpha \in \Omega} \|\alpha\|_1 \leq n$, we get

$$\begin{aligned} &\left| \frac{g_q^{(3)}(\lambda_0, \lambda_1)}{g_q^{(0)}(\lambda_0, \lambda_1)} - \frac{g_q^{(2)}(\lambda_0, \lambda_1) g_q^{(1)}(\lambda_0, \lambda_1)}{g_q^{(0)}(\lambda_0, \lambda_1)^2} \right| \\ &= \left| \mathbb{E}_{p_{\lambda_0}}[\langle \lambda_1, \alpha \rangle^3] - \mathbb{E}_{p_{\lambda_0}}[\langle \lambda_1, \alpha \rangle^2] \mathbb{E}_{p_{\lambda_0}}[\langle \lambda_1, \alpha \rangle] \right| \\ &\leq 2n \left(\mathbb{E}_{p_{\lambda_0}}[\langle \lambda_1, \alpha \rangle^2] - \mathbb{E}_{p_{\lambda_0}}[\langle \lambda_1, \alpha \rangle]^2 \right) \\ &= 2n \cdot D^2 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1]. \end{aligned} \tag{4}$$

Combining 3 and 4 using the triangle inequality, we get that

$$\left| D^3 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1, \lambda_1] \right| \leq 4n D^2 h_{\theta,q}(\lambda_0)[\lambda_1, \lambda_1].$$

Therefore, $h_{\theta,q}$ is $4n$-second order robust. □

References

Allen Zhu, Z., Li, Y., Oliveira, R., and Wigderson, A. Much faster algorithms for matrix scaling. In *FOCS'17: Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, 2017.

Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., and Varshney, K. R. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, pp. 3992–4001, 2017.

Cohen, M. B., Madry, A., Tsipras, D., and Vladu, A. Matrix scaling and balancing via box constrained Newton's method and interior point methods. In *FOCS'17: Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, 2017.

Jerrum, M. R., Valiant, L. G., and Vazirani, V. V. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

Karmarkar, N. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pp. 302–311, 1984.

Singh, M. and Vishnoi, N. K. Entropy, optimization and counting. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pp. 50–59. ACM, 2014.

Wright, M. The interior-point revolution in optimization: history, recent developments, and lasting consequences. *Bulletin of the American mathematical society*, 42(1): 39–56, 2005.

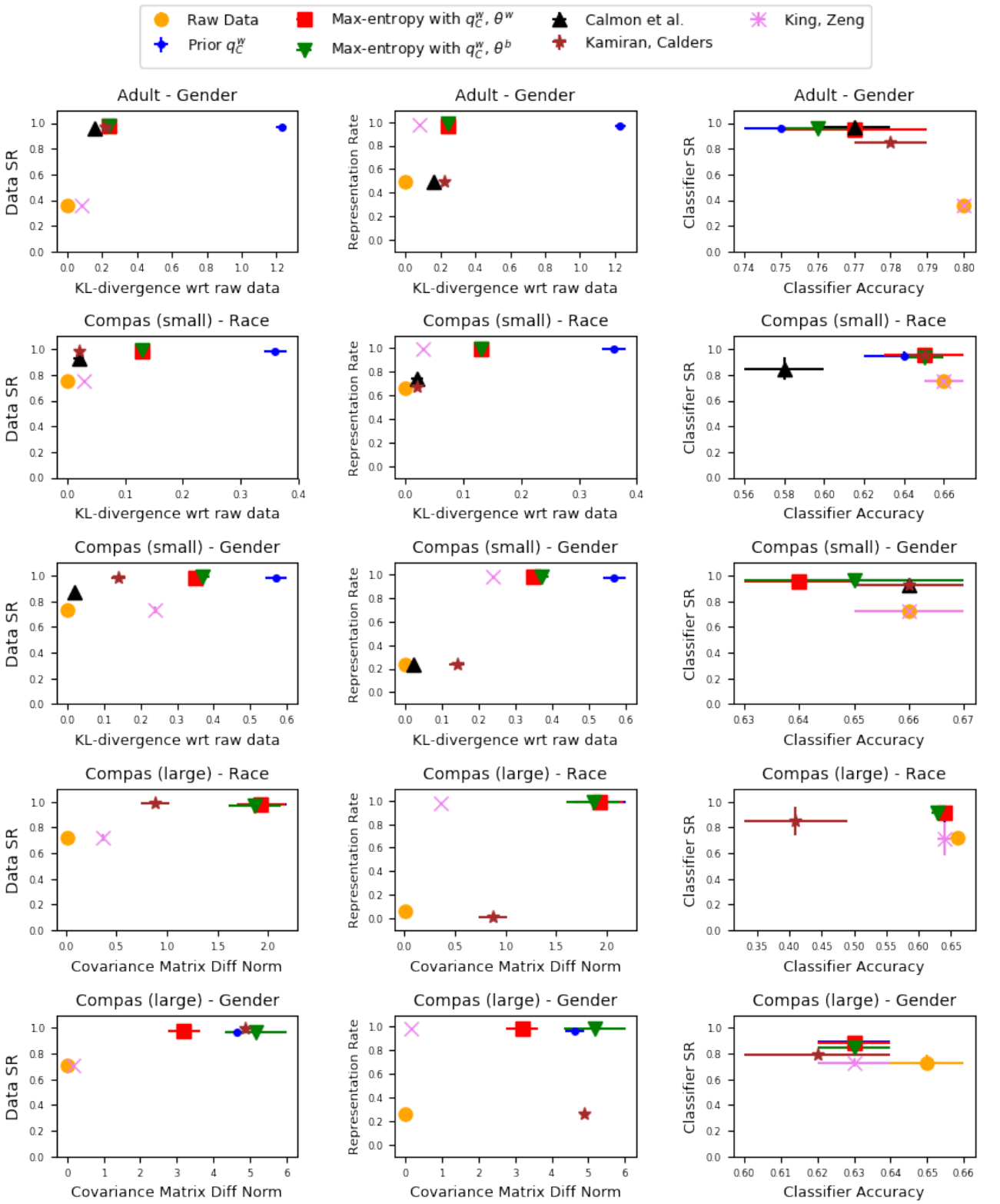


Figure 7. The figures represent the fairness (measured using data SR or classifier SR or representation rate) vs accuracy (measured using KL-divergence or covariance matrix difference norm or classifier accuracy) tradeoff for our method and baselines. "SR" denotes statistical rate. For all metrics, we plot the mean across all folds and repetitions, with the standard deviation as error bars. Note that the approach of (Calmon et al., 2017) is infeasible for larger domains, such as the large version of COMPAS datasets, and hence we do not present their results on that dataset. These results are also presented in tabular form in Table 2 in the supplementary material.